Consistent Re-clustering in Mobile Ad Hoc Networks

C. Tselikis, C. Douligeris, S. Mitropoulos
Dept. of Informatics
University of Piraeus
Piraeus, Greece
{ctselik, cdoulig, sarandis}@unipi.gr

N. Komninos
Algorithms and Security Group
Athens Information Technology
Athens, Greece
nkom@ait.edu

***Abstract*—In this paper we revisit some considerations relative to the performance of re-clustering algorithms in MANET. We recommend that for a secure MANET the design of re-clustering algorithms should not only provide for clustering stability, but also for network robustness in terms of network connectivity, of message reliability, of tolerance against the attacks that target the cluster head nodes and of tolerance to random node failures due to energy drains. We also take into account the possibility of malicious users that might thwart the network protocol by advertising false topology information. We propose a distributed mechanism that for unbiased cluster head election first demands certain levels of consistency to be reached among the nodes.**

I. INTRODUCTION

In mobile ad hoc networks (MANET) the moving nodes communicate in a peer to peer fashion, without the need for a network infrastructure. Re-clustering mechanisms, when used, intend to insert a kind of infrastructure functionality in the ad hoc network. Efficient re-clustering schemes produce stable hierarchical structures that are built over dynamically changing cluster heads which perform multi-hop forwarding, efficient radio channel allocation, intrusion detection, key and cluster membership management. The benefit of clustered ad hoc networks is more evident when their scale is large.

However, re-clustering inserts complexity which some times, in order to sustain the structure, might lead to unacceptable management cost [1]. There might be user mobility patterns, such as far migrations, that could reduce the advantage of using clusters by requiring frequent updates of the routing information. Indeed, topological parameters are factors that challenge the advantage of using clusters in multi-hop packet forwarding. Moreover, the potential of an attack should not be excluded when choosing to re-cluster in MANET. For example, in the case when malicious nodes advertise misleading information pertinent to their connectivity level, or in the case of a DoS attack in which an attacker selectively floods the selected cluster head set, the network performance, connectivity and hence availability are put in threat.

The re-clustering mechanism, wherever present, should be designed adaptable both to targeted attack scenarios and random energy outages so that the communications continue with the least possible message losses.

To this end, we propose a robust re-clustering algorithm which in order to select a cluster head examines the node's degree, which is the number of its one hop neighbors, and also takes into account the node's residual energy. In this way robustness is secured in two ways. The node degree is a dynamic property of the network topology, which is advertised among the neighbors, while the residual energy represents an individual autonomous property of the nodes. A chosen cluster head that lacks those two properties (connectivity and energy) soon or later will cause disruption in its routing paths. In addition, we attempt to overcome the biased cluster head selections by demanding certain levels of consistency among the node advertisements in order to protect from malicious users that thwart the network protocol by advertising false (topology) information.

II. RELATED WORK

In the related work the re-clustering algorithms are either topology-aware (dynamic) or they base the decisions for cluster head selection on static priorities. Moreover, they can be autonomous in the sense that the nodes decide on their own if they are going to announce themselves as eligible cluster head nodes or not.

The Lower ID (LID, hierarchical or not) algorithm [2] is one of the most popular static algorithms with numerous variants. In LID, the cluster head role is assigned to the node with the lowest ID in its vicinity. In case that a highly mobile node with a low ID roams the whole network area, it will prevail in all the visited clusters and this will cause the node's battery exhaustion, numerous re-clustering updates and bandwidth waste. Improved LID schemes include the Least Cluster Head Change algorithm (LCC) [4] and the weight-based Distributed Clustering Algorithm (DCA) [5].

The most popular topology-aware algorithm is the Highest-Degree (HD) algorithm in which the node with the largest number of one-hop neighbors is selected as cluster head. It is obvious that the highest-degree selection criterion is dynamic

and depends on the node's mobility pattern since link failures and link re-connections are caused by the movement of the nodes. Improved schemes of the highest-degree algorithm include the Least Cluster Head Change algorithm [4] and the weighted Highest Degree [3]. In [3] when two cluster heads come in range they both give up roles and the initiated re-clustering procedure yields only one new cluster head rather than two. In [6] a re-clustering protocol is proposed where a weighted metric that combines system-wide parameters is used for the cluster head selection.

The LEACH cluster head selection algorithm [7] mainly applied to sensors networks is the most representative of the autonomous, energy-efficient family of clustering schemes. In [7] the sensors take autonomous decisions to protect themselves from outages and randomization is introduced in order to spread uniformly the total energy available.

III. RE-CLUSTERING ISSUES

In the following we present our primary concerns in the design of an efficient re-clustering scheme, suitable for the infrastructure-less MANET environment.

A. Distributed vs. Centralized Re-Clustering

According to the distributed clustering principle, during the re-clustering procedure the decision-making is based on local information, i.e., the node priorities for cluster head selection are calculated from information gathered from each node's vicinity. On the contrary, in centralized approaches the cluster heads are selected after examining global information, a spendthrift tactic for both the network processors and the available network bandwidth.

B. The Robustness – Stability trade-off

Many trade-offs have already been identified and extensively studied in the constrained wireless ad hoc environment. The security vs. energy consumption and the more general network capacity vs. interference trade off are the most known among others. The trade-off that we identify and try to address in this paper is between connectivity-related robustness metrics, such as the message reliability in intra-cluster and inter-cluster communications, vs. the stability of the different underlying re-clustering schemes. The re-clustering stability criterion is the cluster head change rate which relates to the communications overhead that is imposed by the underlying re-clustering scheme and which affects the network throughput. We will see in the Evaluation Section that the proposed re-clustering algorithm, compared to two other re-clustering algorithms, achieves better connectivity and hence better end to end message reliability but the penalty paid is an increased number of cluster head modifications, which however was evident only in the case of small range radio transmission.

C. Attack-aware Re-Clustering

Examining the security aspect of distributed vs. the centralized systems it is remarkable that centrally controlled networks can be attacked more easily with a single point of failure, while in distributed systems the impact of an attack might be restricted at the local level only without necessarily causing a network disruption. Moreover, reaction to attacks is achieved better when re-clustering is distributed, since the overhead for applying secure re-clustering criteria on messages coming from the node's vicinity is less than that when filtering the messages coming from the entire network area.

Further, the security mechanisms should take into account the possibility of existing compromised nodes inside the network that, for their own interest, might advertise fraudulent information. The malicious users might bias the re-clustering decision procedure by advertising to the network misleading information regarding, for example, the number of neighbors that they have in range, or their energy resources, or their geographical position.

D. Consistent Re-Clustering

The detection of malicious users in MANET with infrastructure has been addressed in pevious works like in [11], however in the absence of both digital signatures and certificates one way to protect the infra-structure-less MANET from malicious users is via distributed mechanisms that perform in-network data processing. We are developing a re-clustering scheme that takes into account the potential of malicious users who send faulty messages on purpose, a condition also addressed in [12]. To this end, we incorporate into the re-clustering procedure a light distributed mechanism that requires certain levels of consistency to exist amongst the claims that are collected from the one-hop cluster head candidates. The resulting clusters are two hop in their diameter.

IV. THE RRA

The re-clustering algorithm that will be evaluated in section VI is a weighted graph-based re-clustering algorithm, the Robust Reclustering Algorithm (RRA). The decision parameters that RRA takes into account in its cluster head selection procedure are:

- The node degree d_i, i.e., the number of one-hop neighbors that each node has inside its radio range.
- The node residual energy, E_{ri}. In general, as for cluster heads preferable are those nodes with sufficient battery resources to perform the cluster head tasks.

The RRA decision variable w_i is the weighted sum of the decision parameters d_i and E_{ri}, as given in Equation (1).

$$w_i = \mathrm{a} \times d_i + b \times E_{ri}, \quad (1)$$

where a, b are the decision coefficients bounded by:

$$a + b = 1$$

The choice of the decision coefficients *a*, *b* depends on the specific ad hoc network configuration and the application's requirements. For example, if we weigh more the energy coefficient *b*, RRA will select as cluster heads the nodes with more energy left, a choice more suitable for sparse networks, while if we weigh more the connectivity degree *a,* the nodes having a large number of one-hop neighbors will be selected, a choice more suitable for dense networks.

RRA consists of two phases. In the network set-up *PHASE I* the nodes are deployed in the field and their roles, energy, counters and routing tables are initialized. In *PHASE II*, the procedures of the cluster head selection and the update of the routing tables are performed.

Network set up procedure

```
∀node ∈G {
        node.Role = Isolated;
        node.CHcounter , node.Econsumed = 0;
         node.Eresidual = max;
         node.CH_Table = new Table;
         node.Members_Vector = new Vector;
         node.Neighbor_List = new List;
         node.Hellos_Vector= new Vector;
      }
```

Fig. 1 The RRA network set-up phase.

Cluster Head selection procedure

```
∀node ∈G {
   //G is the deployed network graph
  if (node ∉D) {
     //D is the set of nodes already joined to clusters
      build Neighbor_List;
      CH = node with MAX wi in Neighbor_List ;
      CH.setRole = cluster_head;
      CH.setCH(CH);
      CH_Table.add(CH);
      CH.Eres = CH.Eres  - CH.Econsumed++;
      ∀(neigh ∈ Neighbor_List and neigh != CH) {
                  neigh.setRole(Member);
                 neigh.setCH(CH);
                 Member_Table.add(neigh);
                 D.add(neigh);
      }
      D.add(node);
      D.add(CH);
    }
  }
```

Fig. 2 The RRA re-clustering phase over the network graph G.

1) PHASE I. In the network set-up phase (Figure 1) the nodes are dispersed across the covered area according to a specific deployment pattern. During the set-up phase RRA allocates to all nodes a unique ID and the isolated role, i.e., initially the nodes are neither cluster heads nor members to any cluster. Also, each node holds a counter CH_{count} recording the times that it has been elected as cluster head and a counter for the energy consumed $E_{consumed}$ which are both initialiased. The routing table (*CH_Table*), the one-hop neighbor list (*NL*) and the *Hellos_Vector* with all the Hellos that the node receives, are also initialised for each node in *PHASE I*.

2) PHASE II. In the second phase (Figure 2), RRA performs the re-clustering procedure. The *Neighbor_List* for each node is built by reading the node_IDs which are included in the received Hello messages. If no neighbor is heard after a specified time interval, then the node becomes cluster head. Each node in the network reads the connectivity degree d_i and the residual energy E_{ri} which are both included in the HELLO messages that each neighbor *i* broadcasts. The node calculates the decision variable w_i by applying Equation (1) for each one of its neighbors. The neighbor that gives the maximum value of w_i will be selected as CH. To associate with a new CH, the node sends a JOIN request message and if the request is acknowledged, the node updates its CH. Also, the CH adds the node's ID to the *Members_Vector* which holds all its cluster members.

We are also developing an enhanced version of the RRA with the aim to protect the cluster head selection procedure from being biased by malicious users who might advertise fraudulent information about their connectivity degree in their vicinity and, in effect, might degrade the ad hoc routing protocol performance. The specification of the Enhanced-RRA is as follows.

The ERRA messages shown in Table I include the JOIN message (a member sends a JOIN to associate with a cluster head), the LEAVE message (sent to disaccosiate a member from its cluster head ID) the ACK message to acknowledge both the JOIN and LEAVE requests of the members and the broadcast HELLO message.

TABLE I. MESSAGE REPERTOIRE OF ERRA

Message	Parameters
JOIN	(node_ID, CH_ID)
LEAVE	(node_ID, CH_ID)
ACK	(node_ID, CH_ID)
HELLO	(node_ID, CH_ID, NL, *d*, E_{res})

The ERRA algorithm (Figure 3) performs the CH selection in a more consistent manner than RRA does, by first sorting the node's neighbors list with decreasing order of the value of the decision variable w_i which is calculated by Equation (1). According to ERRA, the node picks the node ID with the maximum value of w_i as for the first cluster head candidate in the cluster. In ERRA the honesty of the candidates is of primal concern and for this reason the candidate node will be further examined by checking the advertised degree d_i against two thresholds.

This work has been partially funded by the University of Piraeus Research Centre and by a grant from PENED Program of the Greek Research and Technology Secretariat.

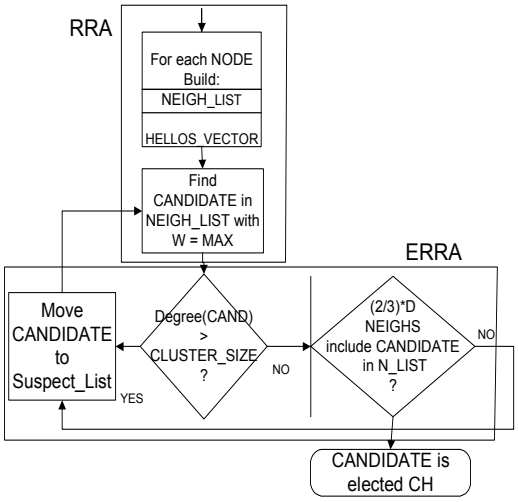

Fig. 3 RRA and ERRA cluster head selection procedures.

a) Regarding the first threshold, the ERRA selection procedure sets a maximum to the number of neighbors that a CH is allowed to advertise [13]. An upper threshold acceptable as for the claimed node degree could be equal to the cluster's size. The cluster size might be, for example, 15 nodes in a total of 200 nodes. In any case, the cluster size should be known only to the cluster heads and not be sent to the members. If the candidate node advertised a connectivity degree d_i which is larger than the upper threshold the candidate is suspected as malicious and is put to the nodes' suspect list. The re-clustering process repeats setting as next candidate the node with the next larger value of the decision variable w_i in the node's *Neighbors_List*. Otherwise, the first candidate node remains as candidate and is examined against the second ERRA threshold.

b) Regarding the second threshold, ERRA requires more than (2/3) * d_i of the one-hop node's neighbors to include in their one-hop Neighbors List (the NL is included in the HELLO packets, as shown in TABLE I) the candidate node ID which declared the maximum degree d_i. This is the necessary and sufficient condition for the existence of distributed consensus [14]. This condition is applied here to reach consensus amongst the neighbors regarding their cluster head election. In other words, ERRA demands the minimum level of consistency between the claim of the primal cluster head candidate and the neighbors who sent HELLOs with the primal candidate included in their *Neighbor_List*.

If consistency exists, the candidate node is elected as CH. If the new cluster head is different than the previous one (for example, the new cluster head might be a visitor node that came into the cluster area), then the necessary re-clustering procedures are initiated, i.e., the cluster members update their CH and the new CH is added to the routing tables. Also, the new CH node switches state, while the old CH gives up his role and turns to simple member. If no consistency (agreement) exists, ERRA aborts the candidate node, moves it to the suspect list and repeats by setting as next cluster head candidate the ID found in the node's sorted NL with the next largest value of the decision variable w_i.

V. SIMULATION PARAMETERS

We conducted simulation experiments by using the JNS simulator [8] with the objective to evaluate the proposed algorithm (RRA) against two well known re-clustering algorithms, namely a static one, the Lower ID (LID), and a topology-aware one, the Highest Degree (HD). We were interested in taking into account different possible network conditions in order to analyze the behavior of the algorithms under these different conditions. To this end, the input parameters that we varied in the simulation were the network density, the initial node deployment pattern over the covered area, the user mobility pattern, the radio transmission range and the packet transmission rate, which was fixed, as shown in Table II.

TABLE II. SIMULATION PARAMETERS

Parameter	Value
Nodes	100 - 200
Surface	1000pel x1000pel
Deployment pattern	Random, Heavy Tail
Network PDU	1024 KBytes
Packet rate	100 packets / sec
Simulation time	4000 secs

1) Mobility Model

In the experiments the mobility was simulated with the Random Waypoint model. According to the random waypoint movement, a node chooses at random a destination point inside the bounds of the network area and with a randomly chosen velocity travels towards this point following a straight path. When the node reaches his destination, pauses for a random time, and then randomly chooses a new destination point and the procedure repeats. We assumed zero pausing times, which corresponds to continuous movement. The nodes were simulated to travel with an average speed in the range between the low pedestrian speed of 5km/h and the high vehicular speed of 50km/h.

2) Node Deployment Model

During the RRA *PHASE I* we set up the MANET by deploying the nodes both randomly and in groups by controlling the node degree distribution. In the random deployment model the node degrees followed the exponential distribution. In the grouped ad hoc deployment in order to skew the degree distribution and generate a more concentrated initial ad hoc configuration the node degrees followed the Heavy Tail Pareto distribution. The topology generator that we used was the BRITE [9]. In all cases the BRITE generator was set to create graphs of an average node

degree equal to 5. Also we let the network to expand according to the Barabasi-Albert linear preferential connectivity model [10] which we applied at the router level.

3) Radio Coverage

The transmission parameter that mostly correlates to the node's connectivity degree is the transmission range of the wireless node, equivalently the transmission power. In the simulation experiments we varied the radio range from 0.5 meter to 200 meters. When nodes transmit in the low range the network is partitioned to many clusters with minimum overlapping among them, while when nodes cover larger areas we obtain a smaller number of significantly overlapping clusters with large membership.

VI. EVALUATION

A. Comparison of Cluster Head Change Rate at 100 nodes

Figure 4 shows the experimental results concerning the CH modification rates for LID, HD and RRA when 100 nodes were placed randomly in the field (notation –R).

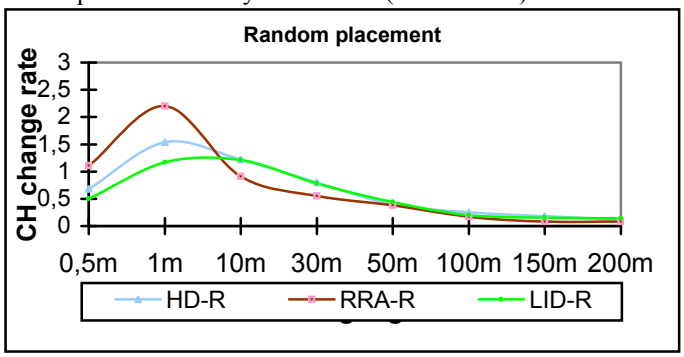

Fig. 4 Comparison of cluster head change rate: 100Nodes, 50km/h.

We observe in Figure 4 that for the sparse network case of 100 nodes with high user speeds of 50km/h, in radio ranges less than 8meters the most stable re-clustering of the three compared algorithms was the LID. On the contrary, when increasing the radio coverage the situation was reversed so that in ranges of more than 8meters the most stable re-clustering was achieved by RRA. In its best performance RRA gave 13.8% less CH changes than the Highest Degree and 13% less changes than the Lower ID.

B. Comparison of the Number of Clusters

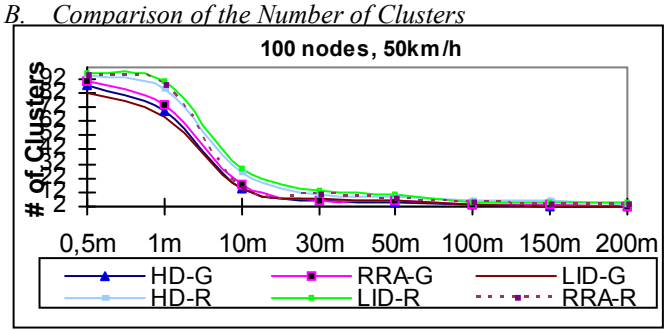

Fig. 5 Comparison of the created clusters: 100Nodes, 50km/h.

Figure 5 compares the number of clusters that were created by the algorithms for different node placements. In the sparse network case of 100 nodes configured in groups (notation –G) fewer clusters were generated than when the same number of nodes was placed randomly in the same field (notation –R). RRA and Highest Degree were proved the most efficient with respect to this metric creating considerably smaller number of clusters when the radio transmission covered 30 meters and more.

C. Comparison of Cluster Head Change Rate at 200 nodes

Figure 6 shows the CH change rates in the case of the dense network with 220 nodes, 5km/h user speed and random placement. The instability point occurred at the 10m radio range for the three algorithms, unlike in the sparser case where the maximum change rate occurred around the one meter radio range. This shift is expected since in dense networks the nodes cross the cluster areas more frequently so that the transition rate between the CH and the member states, in the mean, is larger than it is in the case of sparse networks. In effect, more power is needed for the same level of stability.

RRA was more stable than the rest two algorithms in medium and large radio ranges, as shown in Figure 6. On the contrary, for short to medium ranges the RRA-R overhead was proved greater than the LID-R and HD-R re-clustering overhead. That means that with RRA-R more transmission power is needed to equalize the re-clustering communications overhead in ranges of up to 30m. In the large ranges, HD-R improved performance and efficiently combated RRA-R.

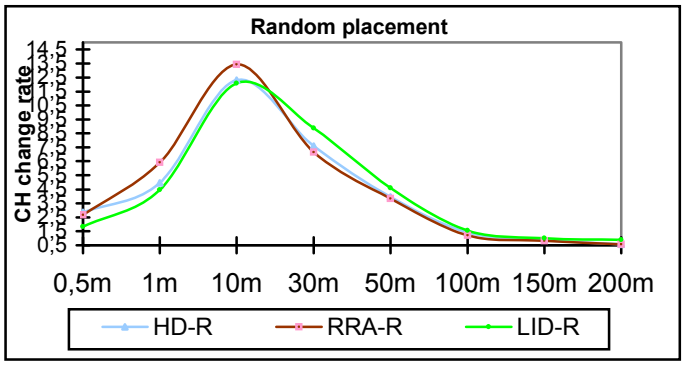

Fig. 6 Comparison of cluster head change rate: 200Nodes, 5km/h.

D. Comparison of Message Reliability and Availability

For our evaluation purposes, the message delivery ratios were calculated, after simulating mixed intra-cluster traffic over the connection-oriented SimpleGoBackN (SGN) and the connection-less UDP transport protocols. The routing protocol was the IP running over the three underlying re-clustering schemes. Figure 7 shows that, with random node placement, RRA was more reliable with respect to the end-to-end message delivery ratio than both the LID and the HD at the average speed of 50km/h. In the dense network case RRA outperformed the LID by a percentage of 10% and was by 2.2% more reliable than HD. LID achieved the worst delivery performance in both the sparse and dense network case, since LID is a static algorithm that doesn't take into account the dynamic changes in the network topology

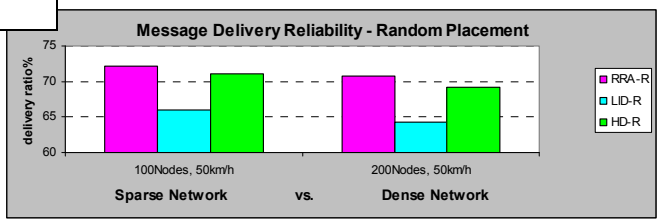

Fig. 7 Message reliability: A mean comparison of the re-clustering algorithms for high speed nodes in the sparse and dense network cases.

Figure 8 illustrates the percentage of the CH nodes that were found available when re-clustering with RRA, LID and HD in the grouped node deployment pattern. The RRA curve demonstrates the earliest rise and the largest percentages of the node availability metric than both LID and HD.

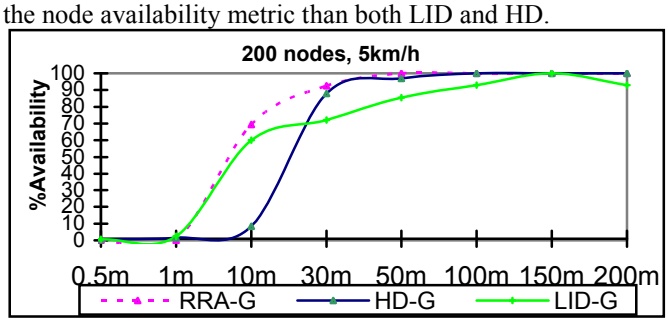

Fig. 8 CH node availability: A mean comparison of re-clustering algorithms in low user speed and grouped deployment.

VII. CONCLUSION

In this paper we presented two algorithms, namely the RRA which is a robust re-clustering algorithm that was specified and evaluated against two traditional re-clustering algorithms and the E-RRA which performs a consistent cluster head election at the local level with the aim to protect the infrastructure-less ad hoc network from fraudulent users. The ERRA specification was supplied in detail.

The random waypoint mobility, the random and the grouped node deployment models were simulated in the experiments. RRA was more stable and created less CH changes than both Lower ID and Highest Degree when the radio transmission of the wireless nodes covered long distances. Moreover, RRA delivered the application level messages to the final destinations more reliably than both the Lower ID and the Highest Degree and also gave the best node availability performance, an index of the RRA robustness against the event of random energy outages and targeted attacks.

However, the trade-off for RRA was to generate high CH modification rates in the case of small radio ranges. This stability degradation of RRA in low transmission power was due to our choice of a and b, the RRA coefficients (relative weights) which form the decision parameter w_i, Equation (1). In more detail, in the experiments we favored as for cluster heads the nodes having larger degree d_i rather than those nodes with more energy left, by choosing a larger value for the coefficient a than the value of the coefficient b. This choice favored the powerful nodes as for CH and hence RRA generated less cluster head changes in the case of large radio ranges, while in the sparse coverage case, where energy is of more importance, RRA degraded in stability.

In the future work we will examine the ERRA consistency condition as for the CH selection in the case of using tables with two-hop neighbor information. Moreover, simulation of targeted attack scenarios, like DoS attacks and routing attacks, is mandated for attack tolerance evaluation of the ERRA versus other known in the literature re-clustering algorithms.

REFERENCES

[1] Akyildiz, I. F., Wang, X. "A Survey on Wireless Mesh Networks". IEEE Radio Communications, September 2005, pp 23-30.

[2] Ephremides, A., Anthony, J., E., Baker, D., J. *A design concept for reliable mobile radio networks with frequency hopping signalling*. Proc. IEEE 1987; vol. 75, no. 1, pp 56-73.

[3] Taniguchi, H., Inoue, M., Masuzawa, T., and Fujiwara, H. "Clustering Algorithms in Ad Hoc Networks". Electronics and Communications in Japan, Part 2, Vol. 88, No. 1, 2005, pp 51-59.

[4] Chiang C. C., Wu, H. K., Liu, W., Gerla, M. "Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel". Proc IEEE SICON, Singapore, 1996.

[5] Basagni, S. *Distributed clustering for ad hoc networks*. Proceedings of the International Symposium on Parallel Architectures, Algorithms and Networks, (I-SPAN'99), Perth/Fremantle. June 1999, pp. 310-315.

[6] El-Bazzal, Z., Kadoch, M., Agba, B. L., Gagnon, F., and Bennani, M. *An Efficient Management Algorithm for Clustering in Mobile Ad Hoc Network.* PM2HW2N'06 Torremolinos, Malaga, Spain, October 6, 2006.

[7] Chandrakasan, A., Heinzelman, W., R., Balakrishnan, H. *Energy-efficient communication protocol for wireless micro sensor networks*. In: 33rd annual Hawaii International Conference on System Sciences, HICSS, page 30053014, 2000.

[8] Java Network Simulator (JNS), <*http://jns.sourceforge.net*>.

[9] Medina, A., Matta, I., and Byers, J. *BRITE: A flexible generator of internet topologies*. Technical Report 2000-005, CS Department, Boston University, Jan. 21, 2000.

[10] Barabasi, A., L., Albert, R. *Emergence of scaling in random networks*. Science, 286(5439), pp 509-512, Oct. 1999.

[11] Komninos, N., Vergados, D., Douligeris, C. *Detecting unauthorized and compromised nodes in mobile ad hoc networks*. Ad Hoc Networks, vol. 5, 2007, pp. 289-298.

[12] Sirivianos, M., Westhoff, D., Armknecht, F., Girao, J. *Non-Manipulable Aggregator Node Election Protocols for Wireless Modelling and Optimization in Mobile Ad Hoc and Wireless Networks,* (ICST WiOpt), 2007.

[13] Karlof, C., Wagner, D. *Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures*, Proceedings of the IEEE International Workshop on Sensor Network Protocols and Applications, May 2003, pp. 113-127.

[14] Fischer, M., Lynch, A., Paterson, M.S. *Impossibility of Distributed Consensus with One Faulty Process.* Journal of the Association for Computing Machinery, vol. 32, No 2, 1985, pp 374-382.